Mobile Intel® Pentium® 4 Processor-M Specification Update

Release Date: April 2002

Order Number: 250721-002

The Mobile Intel® Pentium® 4 Processor-M may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are documented in this Specification Update.

CONTENTS

REVISION HISTORY

Date of Revision	Version	Description
March 2002	-001	Initial Release
April 2002	-002	Added errata T26 and T27; Updated Processor Identification Information; Added Specification Changes T1-T3; Added Documentation Changes T5-T7.

PREFACE

This document is an update to the specifications contained in the following document:

- *Mobile Intel® Pentium®4 Processor-M Datasheet* (Order Numbers 250686)
- *Intel® Architecture Software Developer's Manual, Volumes 1, 2, and 3* (Order Numbers 243190, 243191, and 243192, respectively)

It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools. It contains S-Specs, Errata, Documentation Changes, Specification Clarifications and Specification Changes.

Nomenclature

S-Spec Number is a five-digit code used to identify products. Products are differentiated by their unique characteristics, e.g., core speed, L2 cache size, package type, etc. as described in the processor identification information table. Care should be taken to read all notes associated with each S-Spec number.

Errata are design defects or errors. Errata may cause the Intel Pentium 4 processor's behavior to deviate from published specifications. Hardware and software designed to be used with any given processor must assume that all errata documented for that processor are present on all devices unless otherwise noted.

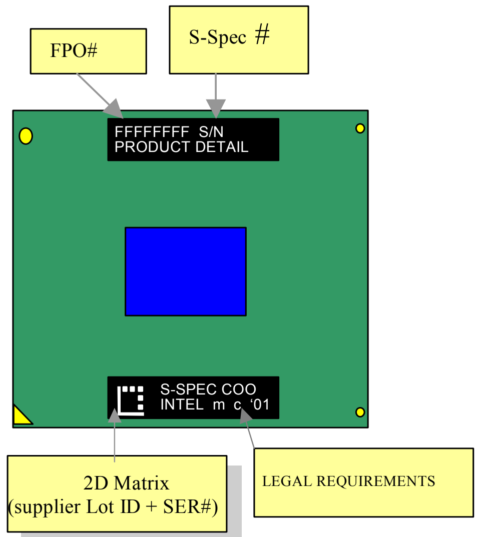
Documentation Changes include typos, errors, or omissions from the current published specifications. These changes will be incorporated in the next release of the specifications.

Specification Clarifications describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in the next release of the specifications.

Specification Changes are modifications to the current published specifications for the Intel Pentium 4 processor. These changes will be incorporated in the next release of the specifications.

GENERAL INFORMATION

Mobile Intel® Pentium® 4 Processor-M (Micro-FCPGA) Markings



IDENTIFICATION INFORMATION

The Mobile Intel® Pentium® 4 Processor-M in Micro-FCPGA package can be identified by the following values:

Family[1]	Model[2]	Brand ID[3]
1111	0010	00001110

NOTES:

1. The Family corresponds to bits [11:8] of the EDX register after Reset, bits [11:8] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the generation field of the Device ID register accessible through Boundary Scan.
2. The Model corresponds to bits [7:4] of the EDX register after Reset, bits [7:4] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the model field of the Device ID register accessible through Boundary Scan.
3. The Brand ID corresponds to bits [7:0] of the EBX register after the CPUID instruction is executed with a 1 in the EAX register.

Mobile Intel® Pentium® 4 Processor-M Identification Information

S-Spec	Product Stepping	L2 Cache Size (bytes)	CPUID	Core Frequency Perf Mode/Bat Mode	Bus Frequency	Voltage Perf Mode/Bat Mode	Package	Notes
SL5YU	B0	512K	0F24h	1.6GHz / 1.2GHz	400MHz	1.3V / 1.2V	Micro-FCPGA	1, 2
SL5Z7	B0	512K	0F24h	1.7GHz / 1.2GHz	400MHz	1.3V / 1.2V	Micro-FCPGA	1, 2

NOTES:

1. V_{CC_CORE} = 1.30V for Maximum Performance Mode; V_{CC_CORE} = 1.20V for Battery Optimized Mode
2. Meets 28W TDP

SUMMARY OF CHANGES

The following table indicates the Errata, Documentation Changes, Specification Clarifications, or Specification Changes that apply to Intel Pentium 4 processors. Intel intends to fix some of the errata in a future stepping of the component, and to account for the other outstanding issues through documentation or specification changes as noted. This table uses the following notations:

CODES USED IN SUMMARY TABLE

X:	Erratum, Documentation Change, Specification Clarification, or Specification Change applies to the given processor stepping.
(No mark) or (blank box):	This item is fixed in or does not apply to the given stepping.
PlanFix:	This erratum may be fixed in a future stepping of the product.
Fixed:	This erratum has been previously fixed.
NoFix:	There are no plans to fix this erratum.
Doc:	Document change or update that will be implemented.
PKG:	This column refers to errata on the Intel® Pentium® 4 processor substrate.
AP:	APIC related erratum.
Shaded:	This item is either new or modified from the previous version of the document.

Each Specification Update item is prefixed with a capital letter to distinguish the product. The key below details the letters that are used in Intel's microprocessor Specification Updates:

A = Intel® Pentium® II processor

B = Mobile Intel® Pentium® II processor

C = Intel® Celeron® processor

D = Intel® Pentium® II Xeon™ processor

E = Intel® Pentium® III processor

G = Intel® Pentium® III Xeon™ processor

H = Mobile Intel® Celeron® processor at 466 MHz, 433 MHz, 400 MHz, 366 MHz, 333 MHz, 300 MHz, and 266 MHz

K = Mobile Intel® Pentium® III processor-M

M = Mobile Intel® Celeron® processor

N = Intel® Pentium® 4 processor

P = Intel® Xeon™ processor

T = Mobile Intel® Pentium® 4 processor-M

The Specification Updates for the Pentium® processor, Pentium® Pro processor, and other Intel products do not use this convention.

Summary of Errata

NO.	B0	Plans	ERRATA
T1	X	NoFix	I/O restart in SMM may fail after simultaneous machine check exception (MCE)
T2	X	NoFix	MCA registers may contain invalid information if RESET# occurs and PWRGOOD is not held asserted
T3	X	NoFix	Transaction is not retried after BINIT#
T4	X	NoFix	Invalid opcode 0FFFh requires a ModRM byte
T5	X	NoFix	FSW may not be completely restored after page fault on FRSTOR or FLDENV instructions
T6	X	NoFix	The processor flags #PF instead of #AC on an unlocked CMPXCHG8B instruction
T7	X	NoFix	When in no-fill mode the memory type of large pages are incorrectly forced to uncacheable
T8	X	NoFix	Processor may hang due to speculative page walks to non-existent system memory
T9	X	NoFix	The IA32_MC1_STATUS register may contain incorrect information for correctable errors
T10	X	NoFix	Debug mechanisms may not function as expected
T11	X	NoFix	Machine check architecture error reporting and recovery may not work as expected
T12	X	NoFix	Cascading of performance counters does not work correctly when forced overflow is enabled
T13	X	NoFix	EMON event counting of x87 loads may not work as expected
T14	X	NoFix	Speculative page fault may cause livelock
T15	X	NoFix	Incorrect data may be returned when page tables are in Write Combining (WC) memory space
T16	X	NoFix	Processor issues inconsistent transaction size attributes for locked operation
T17	X	PlanFix	Multiple accesses to the same S-state L2 cache line and ECC error combination may result in loss of cache coherency
T18		Fixed	Processor may hang when resuming from Deep Sleep state
T19	X	NoFix	When the processor is in the System Management Mode (SMM), debug registers may be fully writeable
T20	X	NoFix	Associated counting logic must be configured when using Event Selection Control (ESCR) MSR
T21	X	NoFix	IA32_MC0_ADDR and IA32_MC0_MISC Registers Will Contain Invalid or Stale Data Following a Data, Address, or Response Parity Error
T22	X	PlanFix	CR2 May Be Incorrect or an Incorrect Page Fault Error Code May Be Pushed Onto Stack After Execution of an LSS Instruction
T23	X	PlanFix	BPM[5:3]# and GHI# V_{IL} Do Not Meet Specification
T24	X	NoFix	Processor May Hang Under Certain Frequencies and 12.5% STPCLK# Duty Cycle
T25	X	NoFix	System May Hang if a Fatal Cache Error Causes Bus Write Line (BWL) Transaction to Occur to the Same Cache Line Address as an Outstanding Bus Read Line (BRL)

Summary of Errata

NO.	B0	Plans	ERRATA
			or Bus Read-Invalidate Line (BRIL)
T26	X	PlanFix	L2 Cache May Contain Stale Data in the Exclusive State
T27	X	PlanFix	Re-mapping the APIC Base Address to a Value Less Than or Equal to 0xDC001000 may Cause IO and Special Cycle Failure

Summary of Documentation Changes

NO.	B0	Plans	DOCUMENTATION CHANGES
T1	X	Doc	Machine Check Exception detected when BINIT# drive enabled
T2	X	Doc	Changes in Volume 2 of the *Intel Architecture Software Developer's Manual*
T3	X	Doc	Performance Counter documentation change
T4	X	Doc	Opcode encodings
T5	X	Doc	Control Registers in Execution Environment
T6	X	Doc	Interrupt 6-Invalid Opcode Exceptions
T7	X	Doc	Interrupt 7-Device Not Available Exception (#NM)

Summary of Specification Clarifications

NO.	B0	Plans	SPECIFICATION CLARIFICATIONS
T1	X	Doc	RCPPS, RCPSS, RSQRTPS, & RSQRTSS Instruction Specification Clarification

Summary of Specification Changes

NO.	B0	Plans	SPECIFICATION CHANGES
T1	X	Doc	Common Clock Output Valid Delay Specification Change
T2	X	Doc	PWRGOOD Specification Changes
T3	X	Doc	PROCHOT# Pulse Width Specification Change

ERRATA

T1. I/O Restart in SMM may Fail after Simultaneous Machine Check Exception (MCE)

Problem: If an I/O instruction (IN, INS, REP INS, OUT, OUTS, or REP OUTS) is being executed, and if the data for this instruction becomes corrupted, the processor will signal a Machine Check Exception (MCE). If the instruction is directed at a device that is powered down, the processor may also receive an assertion of SMI#. Since MCEs have higher priority, the processor will call the MCE handler, and the SMI# assertion will remain pending. However, upon attempting to execute the first instruction of the MCE handler, the SMI# will be recognized and the processor will attempt to execute the SMM handler. If the SMM handler is completed successfully, it will attempt to restart the I/O instruction, but will not have the correct machine state, due to the call to the MCE handler.

Implication: A simultaneous MCE and SMI# assertion may occur for one of the I/O instructions above. The SMM handler may attempt to restart such an I/O instruction, but will have an incorrect state due to the MCE handler call, leading to failure of the restart and shutdown of the processor.

Workaround: If a system implementation must support both SMM and board I/O restart, the first thing the SMM handler code should do is check for a pending MCE. If there is an MCE pending, the SMM handler should immediately exit via an RSM instruction and allow the MCE handler to execute. If there is no MCE pending, the SMM handler may proceed with its normal operation.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T2. MCA Registers may Contain Invalid Information if RESET# Occurs and PWRGOOD is not Held Asserted

Problem: This erratum can occur as a result either of the following events:

- PWRGOOD is de-asserted during a RESET# assertion causing internal glitches that may result in the possibility that the MCA registers latch invalid information.
- Or during a reset sequence if the processor's power remains valid regardless of the state of PWRGOOD, and RESET# is re-asserted before the processor has cleared the MCA registers, the processor will begin the reset process again but may not clear these registers.

Implication: When this erratum occurs, the information in the MCA registers may not be reliable.

Workaround: Ensure that PWRGOOD remains asserted throughout any RESET# assertion and that RESET# is not re-asserted while PWRGOOD is de-asserted.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T3. Transaction is not Retried after BINIT#

Problem: If the first transaction of a locked sequence receives a HITM# and DEFER# during the snoop phase it should be retried and the locked sequence restarted. However, if BINIT# is also asserted during this transaction, it will not be retried.

Implication: When this erratum occurs, locked transactions will unexpectedly not be retried.

Workaround: None identified

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T4. Invalid Opcode 0FFFh Requires a ModRM Byte

Problem: Some invalid opcodes require a ModRM byte (or other following bytes), while others do not. The invalid opcode 0FFFh did not require a ModRM byte in previous generation Intel architecture processors, but does in the Intel® Pentium® 4 processor.

Implication: The use of an invalid opcode 0FFFh without the ModRM byte may result in a page or limit fault on the Intel® Pentium® 4 processor.

Workaround: Use a ModRM byte with invalid 0FFFh opcode.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T5. FSW may not be Completely Restored after Page Fault on FRSTOR or FLDENV Instructions

Problem: If the FPU operating environment or FPU state (operating environment and register stack) being loaded by an FLDENV or FRSTOR instruction wraps around a 64Kbyte or 4Gbyte boundary and a page fault (#PF) or segment limit fault (#GP or #SS) occurs on the instruction near the wrap boundary, the upper byte of the FPU status word (FSW) might not be restored. If the fault handler does not restart program execution at the faulting instruction, stale data may exist in the FSW.

Implication: When this erratum occurs, stale data will exist in the FSW.

Workaround: Ensure that the FPU operating environment and FPU state do not cross 64Kbyte or 4Gbyte boundaries. Alternately, ensure that the page fault handler restarts program execution at the faulting instruction after correcting the paging problem.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T6. The Processor Flags #PF Instead of #AC on an Unlocked CMPXCHG8B Instruction

Problem: If a data page fault (#PF) and alignment check fault (#AC) both occur for an unlocked CMPXCHG8B instruction, then #PF will be flagged.

Implication: Software that depends #AC before #PF will be affected since #PF is flagged in this case.

Workaround: Remove the software's dependency on the fact that #AC has precedence over #PF. Alternately, reload the page in the page fault handler and then restart the faulting instruction.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T7. When in No-Fill Mode the Memory Type of Large Pages are Incorrectly Forced to Uncacheable

Problem: When the processor is operating in No-Fill Mode (CR0.CD=1), the paging hardware incorrectly forces the memory type of large (PSE-4M and PAE-2M) pages to uncacheable (UC) memory type regardless of the MTRR settings. By forcing the memory type of these pages to UC, load operations, which should hit valid data in the L1 cache, are forced to load the data from system memory. Some applications will lose the performance advantage associated with the caching permitted by other memory types.

Implication: This erratum may result in some performance degradation when using no-fill mode with large pages.

Workaround: None identified

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T8. Processor may Hang due to Speculative Page Walks to Non-Existent System Memory

Problem: A load operation that misses the Data Translation Lookaside Buffer (DTLB) will result in a page-walk. If the page-walk loads the Page Directory Entry (PDE) from cacheable memory and that PDE load returns data that points to a valid Page Table Entry (PTE) in uncacheable memory the processor will access the address referenced by the PTE. If the address referenced does not exist the processor will hang with no response from system memory.

Implication: Processor may hang due to speculative page walks to non-existent system memory.

Workaround: Page directories and page tables in UC memory space that are marked valid must point to physical addresses that will return a data response to the processor.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T9. The IA32_MC1_STATUS Register may Contain Incorrect Information for Correctable Errors

Problem: When a speculative load operation hits the L2 cache and receives a correctable error, the IA32_MC1_STATUS register may be updated with incorrect information. The IA32_MC1_STATUS register should not be updated for speculative loads.

Implication: When this erratum occurs, the IA32_MC1_STATUS register will contain incorrect information for correctable errors.

Workaround: None identified

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T10. Debug Mechanisms may not Function as Expected

Problem: Certain debug mechanisms may not function as expected on the processor. The cases are as follows:

- When the following conditions occur: 1) An FLD instruction signals a stack overflow or underflow, 2) the FLD instruction splits a page-boundary or a 64 byte cache line boundary, 3) the instruction matches a Debug Register on the high page or cache line respectively, and 4) the FLD has a stack fault and a memory fault on a split access, the processor will only signal the stack fault and the debug exception will not be taken.
- When a data breakpoint is set on the ninth and/or tenth byte(s) of a floating point store using the Extended Real data type, and an unmasked floating point exception occurs on the store, the break point will not be captured.
- When any instruction has multiple debug register matches, and any one of those debug registers is enabled in DR7, all of the matches should be reported in DR6 when the processor goes to the debug handler. This is not true during a REP instruction. As an example, during execution of a REP MOVSW instruction the first iteration a load matches DR0 and DR2 and sets DR6 as FFFF0FF5h. On a subsequent iteration of the instruction, a load matches only DR0. The DR6 register is expected to still contain FFFF0FF5h, but the processor will update DR6 to FFFF0FF1h.

Implication: Certain debug mechanisms do not function as expected on the processor.

Workaround: None identified

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T11. Machine Check Architecture Error Reporting and Recovery may not Work as Expected

Problem: When the processor detects errors it should attempt to report and/or recover from the error. In the situations described below, the processor does not report and/or recover from the error(s) as intended.

- When a transaction is deferred during the snoop phase and subsequently receives a Hard Failure response, the transaction should be removed from the bus queue so that the processor may proceed. Instead, the transaction is not properly removed from the bus queue, the bus queue is blocked, and the processor will hang.
- When a hardware prefetch results in an uncorrectable tag error in the L2 cache, IA32_MC0_STATUS.UNCOR and IA32_MC0_STATUS.PCC are set but no Machine Check Exception (MCE) is signaled. No data loss or corruption occurs because the data being prefetched has not been used. If the data location with the uncorrectable tag error is subsequently accessed, an MCE will occur. However, upon this MCE, or any other subsequent MCE, the information for that error will not be logged because IA32_MC0_STATUS.UNCOR has already been set and the MCA status registers will not contain information about the error which caused the MCE assertion but instead will contain information about the prefetch error event.
- When the reporting of errors is disabled for Machine Check Architecture (MCA) Bank 2 by setting all IA32_MC2_CTL register bits to 0, uncorrectable errors should be logged in the IA32_MC2_STATUS register but no machine-check exception should be generated. Uncorrectable loads on bank 2, which would normally be logged in the IA32_MC2_STATUS register, are not logged.
- When one half of a 64 byte instruction fetch from the L2 cache has an uncorrectable error and the other 32 byte half of the same fetch from the L2 cache has a correctable error, the processor will attempt to correct

the correctable error but cannot proceed due to the uncorrectable error. When this occurs the processor will hang.

- When an L1 cache parity error occurs, the cache controller logic should write the physical address of the data memory location that produced that error into the IA32_MC1_ADDR register. In some instances of a parity error on a load operation that hits the L1 cache, however, the cache controller logic may write the physical address from a subsequent load or store operation into the IA32_MC1_ADDR register.
- The local xAPIC has an Error Status Register which records all errors which it detects. Bit 6 of this register, the Receive Illegal Vector bit, is set when the local xAPIC detects an illegal vector in a message that it received. When an illegal vector error is received on the same internal clock that the error status register is being written due to a previous error, bit 6 does not get set and illegal vector errors are not flagged.
- If an instruction fetch results in an uncorrectable error and there is also a debug breakpoint at this address, the processor will livelock and the uncorrectable error will not be logged in the machine check registers.
- The MCA Overflow bit should be set when an uncorrectable error resides within the register bank (valid bit is already set) and any subsequent errors occur. The Overflow bit being set indicates that more than one error has occurred. Because of this erratum, if any further errors occur, the MCA Overflow bit will not be updated; thereby incorrectly indicating only one error has been received.

Implication: The processor is unable to correctly report and/or recover from certain errors.

Workaround: None identified

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T12. Cascading of Performance Counters does not work Correctly when Forced Overflow is Enabled

Problem: The performance counters are organized into pairs. When the CASCADE bit of the Counter Configuration Control Register (CCCR) is set, a counter that overflows will continue to count in the other counter of the pair. The FORCE_OVF bit forces the counters to overflow on every non-zero increment. When the FORCE_OVF bit is set, the counter overflow bit will be set but the counter no longer cascades.

Implication: The performance counters do not cascade when the FORCE_OVF bit is set.

Workaround: None identified

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T13. EMON Event Counting of x87 Loads may not Work as Expected

Problem: If a performance counter is set to count x87 loads and floating point exceptions are unmasked, the FPU Operand Data Pointer (FDP) may become corrupted.

Implication: When this erratum occurs, the FPU Operand Data Pointer (FDP) may become corrupted.

Workaround: This erratum will not occur with floating point exceptions masked. If floating point exceptions are unmasked, then performance counting of x87 loads should be disabled.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T14. Speculative Page Fault may cause Livelock

Problem: If the processor detects a page fault which is corrected before the operating system page fault handler can be called e.g. DMA activity modifies the page tables and the corrected page tables are left in a non-accessed or not dirty state, the processor may livelock. Intel has not been able to reproduce this erratum with commercial software.

Implication: Should this erratum be encountered the processor will livelock resulting in a system hang or operating system failure.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T15. Incorrect Data May be Returned When Page Tables Are In Write Combining Memory (WC) Space

Problem: If page directories and/or page tables are located in Write Combining (WC) memory, speculative loads to cacheable memory may complete with incorrect data.

Implication: Cacheable loads to memory mapped using page tables located in write combining memory may return incorrect data. Intel has not been able to reproduce this erratum with commercially available software.

Workarounds: Do not place page directories and/or page tables in WC memory.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T16. Processor Issues Inconsistent Transaction Size Attributes for Locked Operation

Problem: When the processor is in the Page Address Extension (PAE) mode and detects the need to set the Access and/or Dirty bits in the page directory or page table entries, the processor sends an 8 byte load lock onto the system bus. A subsequent 8 byte store unlock is expected, but instead a 4 byte store unlock occurs. Correct data is provided since only the lower bytes change, however external logic monitoring the data transfer may be expecting an 8 byte store unlock.

Implication: No known commercially available chipsets are affected by this erratum.

Workaround: None identified.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T17. Multiple Accesses to the Same S-State L2 Cache Line and ECC Error Combination May Result in Loss of Cache Coherency

Problem: When a Read for Ownership (RFO) cycle has a 64 bit address match with an outstanding read hit on a line in the L2 cache which is in the S-state AND that line contains an ECC error, the processor should recycle the RFO until the ECC error is handled. Due to this erratum, the processor does not recycle the RFO and attempts to service both the RFO and the read hit at the same time.

Implication: When this erratum occurs, cache may become incoherent.

Workaround: None identified.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T18. Processor May Hang When Resuming From Deep Sleep State

Problem: When resuming from the Deep Sleep state the address strobe signals (ADSTB[1:0]#) may become out of phase with respect to the system bus clock (BCLK).

Implication: When this erratum occurs, the processor will hang.

Workaround: The system BIOS should prevent the processor from going to the Deep Sleep state.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T19. When the Processor is in the System Management Mode (SMM), Debug Registers May be Fully Writeable

Problem: When in System Management Mode (SMM), the processor executes code and stores data in the SMRAM space. When the processor is in this mode and writes are made to DR6 and DR7, the processor should block writes to the reserved bit locations. Due to this erratum, the processor may not block these writes. This may result in invalid data in the reserved bit locations.

Implication: Reserved bit locations within DR6 and DR7 may become invalid.

Workaround: Software may perform a read/modify/write when writing to DR6 and DR7 to ensure that the values in the reserved bits are maintained.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T20. Associated Counting Logic Must be Configured When Using Event Selection Control (ESCR) MSR

Problem: ESCR MSRs allow software to select specific events to be counted, with each ESCR usually associated with a pair of performance counters. ESCRs may also be used to qualify the detection of at-retirement events that support precise-event-based sampling (PEBS). A number of performance metrics that support PEBS require a 2nd ESCR to tag uops for the qualification of at-retirement events. (The first ESCR is required to program the at-retirement event.) Counting is enabled via counter configuration control registers (CCCR) while the event count is read from one of the associated counters. When counting logic is configured for the subset of at-retirement events that require a 2nd ESCR to tag uops, at least one of the CCCRs in the same group of the 2nd ESCR must be enabled.

Implication: If no CCCR/counter is enabled in a given group, the ESCR in that group that is programmed for tagging uops will have no effect. Hence a subset of performance metrics that require a 2nd ESCR for tagging uops may result in 0 count.

Workaround: Ensure that at least one CCCR/counter in the same group as the tagging ESCR is enabled for those performance metrics that require two ESCRs and tagging uops for at-retirement counting.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T21. IA32_MC0_ADDR and IA32_MC0_MISC Registers Will Contain Invalid or Stale Data Following a Data, Address, or Response Parity Error

Problem: If the processor experiences a data, address, or response parity error, the ADDRV and MISCV bits of the IA32_MC0_STATUS register are set, but the IA32_MC0_ADDR and IA32_MC0_MISC registers are not loaded with data regarding the error.

Implication: When this erratum occurs, the IA32_MC0_ADDR and IA32_MC0_MISC registers will contain invalid or stale data.

Workaround: Ignore any information in the IA32_MC0_ADDR and IA32_MC0_MISC registers after a data, address or response parity error.

Status: For the steppings affected see the Summary of Changes at the beginning of this section.

T22. CR2 May Be Incorrect or an Incorrect Page Fault Error Code May Be Pushed onto Stack After Execution of an LSS Instruction

Problem: Under certain timing conditions, the internal load of the selector portion of the LSS instruction may complete with potentially incorrect speculative data before the load of the offset portion of the address completes. The incorrect data is corrected before the completion of the LSS instruction but the value of CR2 and the error code pushed on the stack are reflective of the speculative state. Intel has not observed this erratum with commercially available software.

Implication: When this erratum occurs, the contents of CR2 may be off by two, or an incorrect page fault error code may be pushed onto the stack.

Workaround: It is possible for the BIOS to contain a workaround for this erratum.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T23. BPM[5:3]# and GHI# V_{IL} Does Not Meet Specification

Problem: The V_{IL} for BPM[5:3]# and GHI# is specified as 0.9 * GTLREF [V]. Due to this erratum the V_{IL} for these signals is 0.9 * GTLREF - .075 [V].

Implication: The processor requires a lower input voltage than specified to recognize a low voltage on the BPM[5:3]# and GHI# signals.

Workaround: When intending to drive the BPM[5:3]# or GHI# signals low, ensure that the system provides a voltage lower than 0.9 * GTLREF - .075 [V].

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T24. Processor May Hang Under Certain Frequencies and 12.5% STPCLK# Duty Cycle

Problem: If a system de-asserts STPCLK# at a 12.5% duty cycle, the processor is running below 2 GHz, and the processor thermal control circuit (TCC) on-demand clock modulation is active, the processor may hang. This erratum does not occur under the automatic mode of the TCC.

Implication: When this erratum occurs, the processor will hang.

Workaround: If use of the on-demand mode of the processor's TCC is desired in conjunction with STPCLK# modulation, then assure that STPCLK# is not asserted at a 12.5% duty cycle.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T25. System May Hang if a Fatal Cache Error Causes Bus Write Line (BWL) Transaction to Occur to the Same Cache Line Address as an Outstanding Bus Read Line (BRL) or Bus Read-Invalidate Line (BRIL)

Problem: A processor internal cache fatal data ECC error may cause the processor to issue a BWL transaction to the same cache line address as an outstanding BRL or BRIL. As it is not typical behavior for a single processor to have a BWL and a BRL/BRIL concurrently outstanding to the same address, this may represent an unexpected scenario to system logic within the chipset.

Implication: The processor may not be able to fully execute the machine check handler in response to the fatal cache error if system logic does not ensure forward progress on the system bus under this scenario.

Workaround: System logic should ensure completion of the outstanding transactions. Note that during recovery from a fatal data ECC error, memory image coherency of the BWL with respect to BRL/BRIL transactions is not important. Forward progress is the primary requirement.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T26. L2 Cache May Contain Stale Data in the Exclusive State

Problem: If a cacheline (A) is in Modified (M) state in the write-combining (WC) buffers and in the Invalid (I) state in the L2 cache and its adjacent sector (B) is in the Invalid (I) state and the following scenario occurs:

1. A read to B misses in the L2 cache and allocates cacheline B and its associated second-sector pre-fetch into an almost full bus queue,
2. A Bus Read Line (BRL) to cacheline B completes with HIT# and fills data in Shared (S) state,
3. The bus queue full condition causes the prefetch to cacheline A to be cancelled, cacheline A will remain M in the WC buffers and I in the L2 while cacheline B will be in the S state.

Then, if the further conditions occur:

1. Cacheline A is evicted from the WC Buffers to the bus queue which is still almost full,
2. A hardware prefetch Read for Ownership (RFO) to cacheline B, hits the S state in the L2 and observes cacheline A in the I state, allocates both cachelines,
3. An RFO to cacheline A completes before the WC Buffers write modified data back, filling the L2 with stale data,
4. The writeback from the WC Buffers completes leaving stale data, for cacheline A, in the Exclusive (E) state in the L2 cache.

Implication: Stale data may be consumed leading to unpredictable program execution. Intel has not been able to reproduce this erratum with commercial software.

Workaround: It is possible for BIOS to contain a workaround for this erratum.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

T27. Re-mapping the APIC Base Address to a Value Less Than or Equal to 0xDC001000 may Cause IO and Special Cycle Failure

Problem: Remapping the APIC base address from its default can cause conflicts with either I/O or special cycle bus transactions.

Implication: Either I/O or special cycle bus transactions can be redirected to the APIC, instead of appearing on the front-side bus.

Workaround: Use any APIC base addresses above 0xDC001000 as the relocation address.

Status: For the steppings affected see the *Summary of Changes* at the beginning of this section.

DOCUMENTATION CHANGES

The Documentation Changes listed in this section apply to the following documents:

- *Mobile Intel® Pentium® 4 Processor-M Datasheet* (Order Numbers 250686)
- *Intel® Architecture Software Developer's Manual, Volumes 1, 2, and 3* (Order Numbers 243190, 243191, and 243192, respectively)

All Documentation Changes will be incorporated into a future version of the appropriate Mobile Intel Pentium 4 processor-M documentation.

T1. Machine Check Exception Detected When BINIT# Drive Enabled

The last paragraph of section 13.7.1 in *the Intel Architecture Software Developer's Manual, Volume 3: System Programming Guide* currently states:

13.7.1 Machine Check Exception Handler

The MCIP flag in the IA32_MCG_STATUS register indicates whether a machine-check exception was generated. Before returning from the machine-check exception handler, software should clear this flag so that it can be used reliably by an error logging utility. The MCIP flag also detects recursion. The machine-check architecture does not support recursion. When the processor detects machine-check recursion, it enters the shutdown state.

It should state

13.7.1 Machine Check Exception Handler

The MCIP flag in the IA32_MCG_STATUS register indicates whether a machine-check exception was generated. Before returning from the machine-check exception handler, software should clear this flag so that it can be used reliably by an error logging utility. The MCIP flag also detects recursion. The machine-check architecture does not support recursion. When the processor detects machine-check recursion, it enters the shutdown state.

Note: For complete operation of the processors machine check capabilities it is essential that the system BIOS enable BINIT# drive and BINIT# observation. This allows the processor to use BINIT# to clear internal and potentially external blocking states and correctly report a wider range of machine check exceptions. For Example, on an Intel® Pentium® III processor that is executing a locked CMPXCHG8B instruction and a machine check exception is seen on the initial data read, but the comparison operation fails, the processor unlocks the bus after completion of the locked sequence by asserting a BINIT# signal. Without BINIT# drive (UP environment) or BINIT# drive and observation (MP environment) enabled, the machine check error is logged, but the machine check exception is not taken (if MCE's are enabled).

T2. *Changes in Volume 2 of the* Intel Architecture Software Developer's Manual

The *Intel Architecture Software Developer's Manual, Volume 2: Instruction Set Reference* Appendix B, table B-10 is missing the encoding of "Immediate to register" of "AND" instruction. The encoding of "Immediate to register" of "AND" instruction is as follows:

'1000 00sw 11100 reg : immediate data'

The *Intel Architecture Software Developer's Manual, Volume 2: Instruction Set Reference* Appendix B, table B-10 incorrectly documents opcode encoding format of CMPXCHG8B. It currently states:

CMPXCHG8B- Compare and Exchange 8 Bytes
Memory, register 0000 1111 : 1100 0111 : mod reg r/m

It should state:

CMPXCHG8B- Compare and Exchange 8 Bytes
Memory, register 0000 1111 : 1100 0111 : mod 001r/m

The *Intel Architecture Software Developer's Manual, Volume 2: Instruction Set Reference* Appendix B, table B-10 incorrectly documents opcode encoding format of SCASx. It currently states:

SCAS/SCASB/SCASW/SCASD - Scan String 1101 111w

It should state:

SCAS/SCASB/SCASW/SCASD - Scan String 1010 111w

The *Intel Architecture Software Developer's Manual, Volume 2: Instruction Set Reference* Appendix B, table B-10 incorrectly documents opcode encoding format of XCHG. XCHG encoding operand information (1-byte form) does not have a w-bit, hence the reg size is implied. The AL register is not a valid option for this 1-byte encoding. The manual currently states:

XCHG - Exchange Register/Memory with Register

register1 with register2	1000 011w : 11 reg1 reg2
AL, AX, or EAX with reg	1001 0 reg
memory with reg	1000 011w : mod reg r/m

It should state:

XCHG - Exchange Register/Memory with Register

register1 with register2	1000 011w : 11 reg1 reg2
AX or EAX with reg	1001 0 reg
memory with reg	1000 011w : mod reg r/m

T3. Performance Counter Documentation Change

Table A-5, Page A-19 of the Intel Architecture Software Developer's Manual, Vol 3: System Programming Guide, currently states:

Replay metric
2ndL_cache_load_miss_retired

It should state:

Replay metric
2ndL_cache_load_miss_retired(3)

Also add the following note:

3. 2nd-level misses retired does not count all 2nd-level misses. It only includes those references that are found to be misses by the fast detection logic and not those that are later found to be misses.

T4. Opcode Encodings

The *Intel Architecture Software Developer's Manual, Volume 2: Instruction Set Reference* Appendix B incorrectly documents opcode encoding format of several opcodes. Corrections are as follows:

For MOVSS (table B-14),

It currently states:

11110011:00001111:00010000:11 xmmreg1 xmmreg2
11110011:00001111:00010000: mod xmmreg r/m

It should state:

11110011:00001111:00010001:11 xmmreg1 xmmreg2
11110011:00001111:00010001: mod xmmreg r/m

For MULSS (table B-14),

It currently states:

11110011:00001111:010111001:11 xmmreg1 xmmreg2
11110011:00001111:010111001: mod xmmreg r/m

It should state:

11110011:00001111:01011001:11 xmmreg1 xmmreg2
11110011:00001111:01011001: mod xmmreg r/m

For SQRTSS (table B-14),

It currently states:

01010011:00001111:01010001:11 xmmreg1 xmmreg2
01010011:00001111:01010001 mod xmmreg r/m

It should state:

11110011:00001111:01010001:11 xmmreg1 xmmreg2
11110011:00001111:01010001 mod xmmreg r/m

For PMOVMSKB (table B-15),

It currently states:

00001111:11010111:11 mmreg r32

It should state:

00001111:11010111:11 r32 mmreg

For PMOVMSKB (table B-19),

It currently states:

01100110:00001111:11010111:11 xmmreg r32

It should state:

01100110:00001111:11010111:11 r32 xmmreg

T5. Control Registers in Execution Environment

The Intel Architecture Software Developer's Manual, Vol 1: Basic Architecture Section 3.2, under Control register bullet currently states:

> Control registers. The five control registers (CR0 through CR5) determine the operating mode of the processor and the characteristics of the currently executing task (see the section titled “Control Registers” in Chapter 2 of the IA-32 Intel Architecture Software Developer’s Manual, Volume 3).

It should state:

> Control registers. The five control registers (CR0 through CR4) determine the operating mode of the processor and the characteristics of the currently executing task (see the section titled “Control Registers” in Chapter 2 of the IA-32 Intel Architecture Software Developer’s Manual, Volume 3).

T6. Interrupt 6-Invalid Opcode Exceptions

The *Intel Architecture Software Developer's Manual, Vol 3: System Programming Guide* Section 5.14, under fourth and fifth bullets of Interrupt 6-Invalid Opcode Exception (#UD), it currently states:

- Attempted to execute an MMX instruction or an SSE or SSEE2 SIMD instruction (with the exception of the PAUSE, PREFETCHh, SFENCE, LFENCE, MFENCE, and CLFUSH instructions) when the EM flag in control register CR0 is set (1).

- Attempted to execute an SSE or SSE2 instruction when the OSFXSR bit in control register CR4 is clear (0). Note this does not include the following SSE and SSE2 instructions: MASKMOVQ, MOVNTQ, MOVNTDQ, MOVTPD, MOVNTI, PREFETCHh, SFENCE, LFENCE, MFENCE, and CLFLUSH, or the 64-bit versions of the PAVGB, PAVGW, PEXTRW, PINSRW, PMAXSW, PMAXUB, PMINSW, PMINUB, PMOVMSKB, PMULHUW, PSADBW, PSHUFW, PADDQ, and PSUBQ instructions.

It should state:

- Attempted to execute an MMX instruction or an SSE or SSE2 SIMD instruction (with the exception of the MOVNTI, PAUSE, PREFETCHh, SFENCE, LFENCE, MFENCE, and CLFLUSH instructions) when the EM flag in control register CR0 is set (1).
- Attempted to execute an SSE or SSE2 instruction when the OSFXSR bit in control register CR4 is clear (0). Note this does not include the following SSE and SSE2 instructions: MASKMOVQ, MOVNTQ, MOVNTI, PREFETCHh, SFENCE, LFENCE, MFENCE, and CLFLUSH, or the 64-bit versions of the PAVGB, PAVGW, PEXTRW, PINSRW, PMAXSW, PMAXUB, PMINSW, PMINUB, PMOVMSKB, PMULHUW, PSADBW, PSHUFW, PADDQ, and PSUBQ instructions.

T7. Interrupt 7-Device Not Available Exception (#NM)

The *Intel Architecture Software Developer's Manual, Vol 3: System Programming Guide* Section 5.14, under third bullet of Interrupt 7-Device Not Available Exception (#NM), it currently states:

- The processor executed an x87 FPU, MMX, SSE, or SSE2 instruction (with the exception of the PAUSE, PREFETCHh, SFENCE, LFENCE, MFENCE, and CLFLUSH instructions) while the TS flag in control register CR0 was set and the EM flag is clear.

It should state:

- The processor executed an x87 FPU, MMX, SSE, or SSE2 instruction (with the exception of the MOVNTI, PAUSE, PREFETCHh, SFENCE, LFENCE, MFENCE, and CLFLUSH instructions) while the TS flag in control register CR0 was set and the EM flag is clear.

SPECIFICATION CLARIFICATIONS

The Documentation Changes listed in this section apply to the following documents:

- *Mobile Intel® Pentium® 4 Processor-M Datasheet* (Order Numbers 250686)
- *Intel® Architecture Software Developer's Manual, Volumes 1, 2, and 3* (Order Numbers 243190, 243191, and 243192, respectively)

All Specification Clarifications will be incorporated into a future version of the appropriate Mobile Intel Pentium 4 processor-M documentation.

T1. RCPPS, RCPSS, RSQRTPS, and RSQRTSS Instruction Specification Clarification

Volume 2: Instruction Set Reference Manual of the *IA-32 Intel Architecture Software Developer's Manual* (pages 3-665, 3-667, 3-685, 3-687) details the behavior of the RCPPS, RCPSS, RSQRTPS, and RSQRTPS instructions. The documentation for each instruction will be clarified to include the text below about the relative error for the instruction and the values for which flushing results to zero is performed:

> The relative error for this approximation is: |Relative Error| <= $1.5 * 2^{-12}$

For the RCPPS and RCPSS instructions, tiny results are always flushed to 0.0, with the sign of the operand. (Input values greater than or equal to |$1.11111111110100000000000B * 2^{125}$| are guaranteed to not produce tiny results, input values less than or equal to |$1.00000000000110000000001B * 2^{126}$| are guaranteed to produce tiny results, which are in turn flushed to 0.0; and input values in between this range may or may not produce tiny results, depending on the implementation.) Underflow results are always flushed to 0.0, with the sign of the operand.

SPECIFICATION CHANGES

The Documentation Changes listed in this section apply to the following documents:

- *Mobile Intel® Pentium® 4 Processor-M Datasheet* (Order Numbers 250686)
- *Intel® Architecture Software Developer's Manual, Volumes 1, 2, and 3* (Order Numbers 243190, 243191, and 243192, respectively)

All Specification Changes will be incorporated into a future version of the appropriate Mobile Intel Pentium 4 processor-M documentation.

T1. Common Clock Output Valid Delay Specification Change

The timing for Common Clock Output Valid Delay max in Table 19 (System Bus Common Clock AC Specifications) of the Mobile Intel® Pentium® 4 Processor-M datasheet and EMTS Rev 2.0 specification has changed to 1.55 ns from 1.27ns. Designs that follow Intel platform recommendations on the 845 MP chipset will have sufficient margins and will not have any impact.

Table 19. System Bus Common Clock AC Specifications

T# Parameter	Min	Max	Unit	Figure	Notes[1,2,3]
T10: Common Clock Output Valid Delay	0.12	1.55	ns		4
T11: Common Clock Input Setup Time	0.65		ns		5
T12: Common Clock Input Hold Time	0.40		ns		5
T13: RESET# Pulse Width	1	10	ms		6,7,8

Notes:

1. Unless otherwise noted, all specifications in this table apply to all processor frequencies.
2. Not 100% tested. Specified by design characterization.
3. All common clock AC timings for AGTL+ signals are referenced to the Crossing Voltage (V_{CROSS}) of the BCLK[1:0] at rising edge of BCLK0. All common clock AGTL+ signal timings are referenced at GTLREF at the processor core.
4. Valid delay timings for these signals are specified into the test circuit described in Figure 8 and with GTLREF at 2/3 V_{CC} ± 2%.
5. Specification is for a minimum swing defined between AGTL+ V_{IL_MAX} to V_{IH_MIN}. This assumes an edge rate of 0.4 V/ns to 4.0 V/ns.
6. RESET# can be asserted asynchronously, but must be deasserted synchronously.
7. This should be measured after V_{CC} and BCLK[1:0] become stable.
8. Maximum specification applies only while PWRGOOD is asserted.

T2. PWRGOOD Specification Changes

The following changes will be made to the *Mobile Intel® Pentium® 4 Processor-M Datasheet*:

- In the Systems Bus Pin Groups Table 3 PWRGOOD will be moved from the Asynchronous GTL+ Input group to the Power/Other group.
- In the Chapter 2 section titled 'Asynchronous GTL+ Signals' PWRGOOD will be removed from the list of signals.
- In Chapter 5, the Signal Buffer Type of PWRGOOD will change from Asynch GTL+ to Power/Other (in both table 35 and 36).
- Table 14 will be renamed to 'PWRGOOD and TAP Signal Group DC Specifications.'
- TAP will be removed from the descriptions of the VHYS, VT+, and VT- parameters in table 14.
- Table 21 will be renamed to 'Miscellaneous Signals AC Specifications.'
- Timing T35 will have ', except PWRGOOD' removed from the parameter description.
- Note 2 to table 21 will have the following sentence added: 'PWRGOOD is referenced to the BCLK0 rising edge at 0.5 * VCC.'
- Table 28 will be renamed to 'Ringback Specifications for PWRGOOD and TAP Signal Groups.'
- The Signal Groups rows of table 28 will change to read: 'TAP and PWRGOOD.'
- Figure 28 will be renamed to 'Low-to-High System Bus Receiver Ringback Tolerance for PWRGOOD and TAP Buffers.'
- Figure 29 will be renamed to 'High-to-Low System Bus Receiver Ringback Tolerance for PWRGOOD and TAP Buffers.'
- Table 32 will be renamed to 'Asynchronous GTL+, PWRGOOD, and TAP Signal Groups Overshoot/Undershoot Tolerance.'

T3. PROCHOT# Pulse Width Specification Change

The minimum *PROCHOT# Pulse Width Specification (T38) in* Table21 (Asynchronous GTL+ Signals AC Specifications) in the *Mobile Intel® Pentium® 4 Processor-M Datasheet* currently states:

T# Parameter	Min	Max	Unit	Figure	Note
T38: PROCHOT# Pulse Width	TBD		µs	18	5

It should state:

T# Parameter	Min	Max	Unit	Figure	Note
T38: PROCHOT# Pulse Width	500		µs	18	5